Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2007 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2007

A Two-stage Evaluation of User Query Performance for the Relational Model and SQL

Hock Chuan Chan
National University of Singapore, chanhc@comp.nus.edu.sg

112. A Two-stage Evaluation of User Query Performance for the Relational Model and SQL

Hock Chuan Chan
National University of Singapore
chanhc@comp.nus.edu.sg

Abstract
What are the impacts of a data model and a query language on user query performance? This is a longstanding research question about database query. Current knowledge has provided their combined effect. A data model comprises structures and operations, and could be de-coupled from a query language. It is theoretically possible to identify the effects from a data model (without the query language), and the additional effects when a query language is included. An experiment was conducted to provide answers on these effects. Subject query performance with the relational model and SQL was measured at two query stages: the query translation and query writing stages. The experiment confirms literature findings about SQL query difficulties (which are all based on the query writing stage). Exploratory analysis of query difficulties show surprises. For example, operations generally perceived to be difficult (such as joins, group count and repeated relations) are not difficult at the query translation stage, i.e. the difficulties are not because of the relational model, but because of SQL. The study illustrates an approach for separating the effects of data model and query language, which can be used for future studies of other models and languages.

Keywords: user performance, query process, relational model, SQL, data model

Introduction

Databases form an essential component of most management information systems and query languages enable users to directly access essential corporate information. Effects of data models and query languages have been an active area of research (Bowen et al. 2004; Chan et al. 2005; Siau and Tan 2006; Teo et al. 2006) Many empirical studies have been made on the difficulties that users have in using query languages. Some recent examples are: SQL problems through iconic interfaces (Aversano et al., 2002), effects of normalization on end user query (Bowen and Rohde, 2002), effect of ambiguity and ontological clarity on query performance (Borthick et al., 2001; Bowen et al., 2004), effect of data model and languages on query performance (Chan et al., 1999), development of new conceptual query languages (Owei and Navathe, 2001) and natural languages (Owei, 2000; Kang et al., 2002).

A data model comprises a set of structures, constraints and operations (Date, 2001). Batini et al. (1992, p.26) provide this definition: "A data model is a collection of concepts that can be used to describe a set of data and operations to manipulate the data." Operations could be expressed in different languages. For example, relational operations such as join, selection and projection could be expressed in SQL, Query-by-Example, relational calculus or relational algebra. Empirical studies so far have treated the model and query language as an integrated unit (Jih et al., 1989; Chan et al., 1993). Many studies have focused on SQL, which is the dominant query language for the relational database (Date, 1989; Date, 2001).

Many studies have been made to compare different data models and different query languages for their effects on user query performance (Jih et al., 1989; Yen and Scamell,

1993; Chan et al., 1993; Wu et al., 1994; Weber, 1996; Siau et al., 1997; Chan et al. 2005). Query language is an important area of research because it provides access to essential corporate information.

A research question that has remained unsolved is: how much of the query difficulty is due to the model and how much is due to the additional query language syntax? To answer this question, we now measure the effect of the model (including operations but without a query language), as well as the additional effect from a query language (SQL). This is done for queries involving various operations. The results allow us to know which operations are "inherently" difficult as an effect of the model, and which operations are "additionally" difficult because of the SQL syntax. We choose the relational database and SQL because SQL is the industry (and ANSI and ISO) standard query language, and the relational database is still the most widely used database (Ramakrishnan and Gehrke, 2000; Hoffer, 2002). The findings can afford us a better understanding of the user difficulties about the model and language, and can lead us to better focus our user training and database interface design.

Section 2 presents a cognitive model of the query process, which is very relevant for separating the effect of the model from the additional effect of the query language on query performance. Section 3 presents the research model and experiment methodology, together with a brief literature review of research that investigates the difficulty of SQL. Section 4 reports the results of the experiment. The conclusion is given in section 5.

A Cognitive Model of Database Query

This section provides a cognitive perspective on how the data model and query language influence query performance. Ogden (1985) proposes a three-stage cognitive model of database query: query formulation stage (stage 0), query translation stage (stage 1), and query writing stage (stage 2). The model is illustrated in Figure 1. It should be noted that "query writing" is used commonly in the literature to refer to stage 1 and 2 together. This paper follows the tradition for the usage of "query writing", and uses "query writing stage" to refer to this stage of the model.

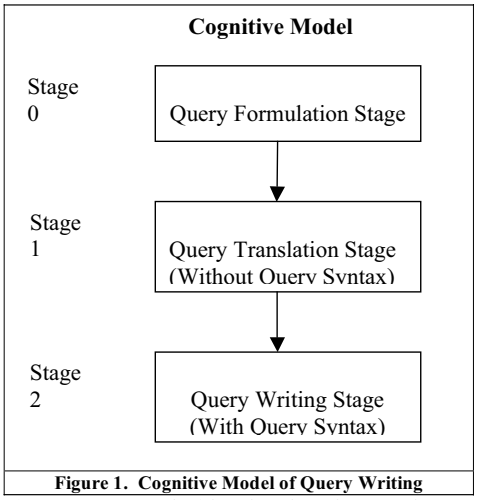

Figure 1. Cognitive Model of Query Writing

For the query formulation stage, users decide what data they need. One example is "I need to know the names of employees who work in the sales department." This stage just uses the knowledge of the application domain. In experiments on query performance, this stage is usually given by the experimenter.

In the query translation stage, users use the output of stage 0 as input, and decide what elements of the data model are relevant, and the necessary operations. One example of the output of this stage is "The employee relation is needed, the column name is to be selected, and a restriction of working in the sales department must be specified on column department, and I need to check with department relation." This output need not be written down. It is usually left in the mind of the users.

In the query writing stage, users arrange the output from stage 1 into the format required by the query language. A simple example, in SQL, is: "select employee.name from employee". This stage is heavily dependent on the particulars of the query language, e.g. the keywords, and the order of the operations and statements.

This cognitive model from Ogden (1985) is consistent with other query models in the literature. For example, Mannino (2001) proposes a similar model, with two steps for users to organize the query syntax. One step is getting from a problem statement to a database representation, which involves a detailed knowledge of the relational tables and relationships. This step is the equivalent of the query formulation stage. The second step is to translate the database representation into the database query language statement, which requires users to develop an allocation of statements for each kind of operator. This is the equivalent of the query writing stage.

Furthermore, Reisner (1977) proposes a similar model. The model states that the user will generate a set of lexical items, which are "created by a (human) process which transforms the English sentence into the relevant query components" (p226), and the user will also identify or generate a query template. The lexical items will then be merged with the template to form the final query. Generation of the lexical items corresponds to the query translation stage – the identification of data structures and operations needed for the query. Generation of the

template and merging it with the lexical items for the final query together correspond to the query writing stage.

This cognitive model has been incorporated in system implementations. The stages can be seen in systems that change a query from one language to another, e.g. in natural language query processing (Androutsopoulos et al., 1995; Galatescu, 2001; Kang et al., 2002), or in mapping an object-oriented query into a relational query (Qian and Raschid, 1995; Wong and Luk, 1996). As proposed in Androutsopoulos (1995) and Kang et al. (2002), a natural language query is changed to a database language query in two stages: first, the question is translated into a meaning representation using linguistic knowledge, and this is then mapped into a formal query.

Another similar model is the human-computer interaction model (Hutchins et al. 1985), which is also used by Liao and Palvia (2000) to explain differences in data models. In this model, there is a distance between the user goal and the user action. This distance includes a semantic distance (e.g. knowing the meaning of the data model relative to the real world) and an articulatory distance (e.g. drawing the data model in the correct shapes and connections). This concept of a semantic distance is also used in the study by Suh and Jenkins (1992). For the query task, going from stage 1 to stage 2 of Ogden's model represents the semantic distance of knowing which parts of the data model are meaningful and relevant for the query. Going from stage 2 to stage 3 of Ogden's model represents the articulatory distance, of articulating the semantics in the proper query language syntax.

Experiments on query performance have measured user performance after stage 2. Thus the findings from the literature cannot separate the impacts of the data model and the query language on the whole query process. If we can measure user performance after stage 1, and after stage 2, it will be possible to have a better understanding of the relative impact of the data model and the language syntax.

Research Model and Methodology

Research hypothesis

The performance of a database user is influenced by four major factors: data-model/query language, task-nature, user and system characteristics (Reisner, 1981; Chan et al., 1993). In this study, user characteristics, which could include age and experience, are controlled through randomizing subjects in an experiment design. System characteristics, which could include user friendliness, dialogue style and system speed, are controlled by having a customized interface designed just for the experiment. The interface presents questions, data and accepts queries in the same way for all the subjects. There are no feedbacks in the interface system.

The task refers to the type of the problem such as data retrieval or data modeling. For this study, task is set at two stages -- the first stage is query translation, i.e., instances are presented directly in the interface, users are asked to select the query result in the interface. Thus we test their understanding of the data value representation of the relational model. The second stage requires the users to write down the query syntax. Thus we test whether they can specify the query operations with SQL syntax. Both stages cover the same query questions.

The data model is defined as having three components: the data model structure, the operations and any constraints on the operations (Codd, 1980; Date, 2001, Batini et al., 1992). The operations could be expressed in different languages. It is controlled by having a customized interface which is designed for the relational data model. The query language refers to the particular formal computer language with all its syntax and semantics, with which a user can express formally the required data and operations. For this study, we include the following operations in the test queries: projection, selection, join, repeated relation, group+count, and not_exist_subquery.

There have already been many empirical studies on SQL and some of these studies indicate the kind of difficulties users face with SQL. For example, Smelcer (1995) reports on the experimental test of several causes of user errors while composing database queries in SQL. His study reveals that join clause omission was a frequent and troublesome error. Miltrovic (1998) also mentions some causes of user query errors of SQL. Some errors come from the burden of having to memorize database schemas; for example, incorrect solution may contain incorrect relational table or attribute names. Other errors come from misconceptions in user's understanding of the elements of SQL and the relational data model in general. He argues that grouping, join conditions and the difference between aggregate and scalar functions are common sources of confusion. Welty and Stemple (1981) found in an experimental study on SQL that users have difficulty with the join and grouping operations in SQL. Borthick et al. (2001) argue that SQL semantic errors include errors such as incorrect use of query operations or operands, missing parts of WHERE conditions, missing table-join conditions, and missing substring functions.

These previous research works point to many SQL difficulties for end-users. According to the two-stage model in figure 1, the query writing stage involving SQL is an additional task after the query translation stage. User query performance is very commonly measured in terms of query accuracy (Chan et al. 1993, Bowen et al. 2004). Since SQL is known to be difficult, we make the following hypothesis:

H1: The query writing stage will have lower query accuracy than the query translation stage.

Research method

A laboratory experiment was conducted to determine the effects of different query operations of SQL on user performance. Twenty subjects participated in the experiment. Each subject performed seven queries for both stages. The queries covered a comprehensive range from the very simple to the very difficult. The 7 chosen queries covered the following semantic specifications: single entity, two entities (of different types) connected by a relationship, attribute condition, two instances of the same type, counting of relationships, quantifiers for where, exist and not exist. These cover the basic queries that are commonly made on the relational model (Connolly and Begg, 2002; Rob and Coronel, 2002). Every query involved different combination of operations and the previous query question had no connection with the following one. Questions are sequenced generally from the easy to the difficult questions, as is done in many query experiments (Jih et al., 1989; Chan et al., 1993). This is confirmed by the actual query performance. Any learning effect will favor the difficult operations, and thus any identification of difficult operations will be on the conservative side.

Variables

The independent variable is the task (query translation and query writing). The dependent variable, performance, is the measure of query accuracy. The accuracy of each answer,

measured as a binary 0 or 1, was determined separately by two graders. A totally correct answer gets a score of 1, otherwise the score is 0. Accuracy is measured for each of the two query stages. The robustness of the findings was also validated with different grading schemes (Chan and Wei, 1996).

Subjects

User characteristics were controlled as follows: 20 first year undergraduate students of a computing faculty participated. The number of subjects is comparable to the number of subjects used in many other similar studies, e.g., 44 (further divided into two groups randomly) in Shoval and Shiran (1997), 10 in Sirinvasan and Irwin (1999), and 66 (further divided into three groups) in Liao and Palvia (2000). All the subjects had used computers before but had no database experience. Subjects were paid for participation as well as for performance.

Training

Subjects were trained by an administrator before they took the query test. A training booklet was used during the experiment. Training booklet gave a brief overview of relational data model and the query language. Subjects practiced answering a question after each example. Feedback on query accuracy was given to improve learning before proceeding to the next example. Training continued until the subjects were fully satisfied. Time of training was about one hour.

Test

After a ten-minute break, the subjects had a practice session so that they could get familiar with the mechanics of the interface. For the test, subjects answered seven questions based on a new database domain. The program displayed the questions one by one. An example of the interface is shown in figure 2. Subjects first finished the query translation task and then the query writing task for each query question.

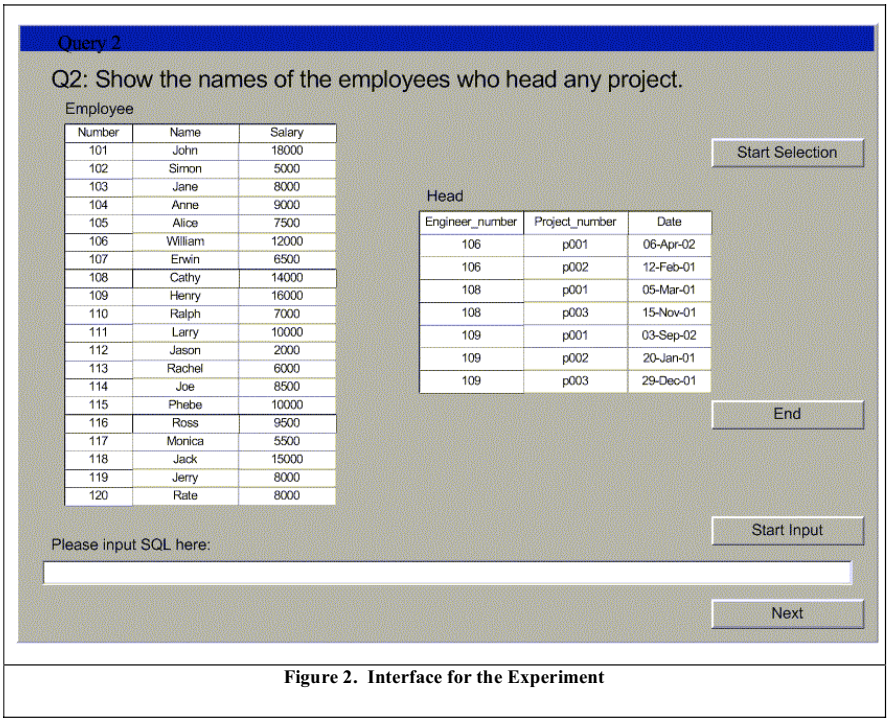

Figure 2. Interface for the Experiment

For the query translation task, subjects have to select answers from the interface where the relation structure and a few data rows are directly represented. Subjects just need to click the data values they want and the system will highlight the selection and record it automatically. This test aims to capture subject performance at stage 1 of Ogden's model. However, there is a slight difference. With Ogden's model, the processing and result of stage 1 could be left in the subject's mind. That is, according to Ogden's model, the subject simply needs to know what to do. In our test, the subject has to demonstrate that he knows what to do, by selecting data results from a few rows of data. Thus, the performance measured by the test at stage 1 is a lower bound estimate of the real query translation performance. The real performance is likely to be better than or equal to the measured performance, for stage 1.

For the query writing stage, subjects have to write down the formal SQL query. The system records the answer, time taken, as well as prompts for the confidence value for each query. For this stage, real performance is likely to be equal to the measured performance, as the test does not introduce any additional steps to Ogden's model. Thus, any difference in the measured performances across the stages is likely to be less than the real difference, and thus making the real difference more statistically significant than reported.

Subjects could refer to the training material and use paper and pencil to help formulate answers. Subjects were given a relational schema of a set of relations, on paper, of the relational model. The test materials, query questions and sample answers can be found in the Appendix.

Hypothesis Test

Two graders separately determined the accuracy of the subjects' answers. Each subject gets seven grades for the seven query questions. Initial grader scores have correlations of above 0.95. Each case of a difference in score was discussed and a common score was given. The mean score (per query) and the standard deviation (in parenthesis) are 0.86 (0.14) for the query translation stage, and 0.44 (0.15) for the query writing stage. Since data on both stages do not follow a normal distribution, non-parametric tests were used, with the SPSS software. "Ordinarily, non-parametric procedures are equivalent to parametric procedures applied to appropriately transformed data" (Rosenthal and Rosnow, 1991, p.316).

The Wilcoxon's Signed Ranks Test is used to compare the data across the two stages, given a z value of -3.862 and p value of 0.001. This means that the query writing accuracy is significantly lower than the query translation accuracy. Thus, hypothesis H1 is supported. In addition to statistical significance, the difference may be considered large – 0.42 on a scale of 0 to 1.

Non-parametric tests between the accuracy values from the two stages for each query show significant p values, from 0.001 to 0.029, except for query 1. The results provide further empirical evidence that the subjects get significantly higher accuracy at the query translation stage for most queries than at the query writing stage. Having to arrange the operations in the SQL syntax significantly deteriorates user query performance.

The results provide an estimate of the difficulty caused by the data model. This is an unavoidable difficulty with the model. It represents the minimum difficulty users will face if they have a perfectly easy query language that does not add any further difficulty. The large gap between the translation stage and the writing stage shows the huge difficulty caused by the SQL language. It will be of interest to examine how other query languages, such as Query-By-Example (QBE), can reduce the gap.

Exploratory Analysis of Operation Effects

We further explored how different operations may lead to different accuracies, and how their effects differ across the query stages. The data are shown in Table 1. The Wilcoxon's Signed Ranks Test is used to compare user query performance of different query questions, within each stage. The difference in performance between 2 queries is attributed to the additional operation in one of the queries. Table 2 shows the comparison results of various operations that are involved in the corresponding query questions. For example, query 2 has a join operation compared to query 1, and both queries have a projection operation. Thus, the difference in performance between query 1 and 2 is attributed to the join operation. Table 2, row 1 shows that the join operation has no significant effect during the query translation stage (Q1 is not found to be significantly different from Q2). For the query writing stage, Q2 is significantly worse than Q1, indicating that the join operation has a significant downward effect on performance. Similarly, Q3 has an extra selection and extra join compared to Q2, and they both have a projection operation and a join operation. The difference in performance between Q2 and Q3 can be attributed to the additional selection and join operation. However, table 2, row 2 shows no significant differences between Q2 and Q3, thus indicating that the selection operation has no significant effect on performance, and further that one additional join does not lead to additional difficulty. This further confirms the findings from comparing

Q1 and Q2. The difference in Q1 and Q4 could be attributed to the additional selection and repeated relations in Q4. However, since we know that selection has no effect, the difference is attributed to repeated relations. The others are deduced similarly.

Table 1. Performance Data for the Queries

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Operations	Proj.	Proj. + join	Proj. + sel. + joins	Proj. + sel. + repeated relation	Proj. + sel. + repeated relation + joins	Proj. + join + group count	Proj. + join + not_exist subquery
Query Translation [1]	19	20	18	16	12	20	15
Query Writing [2]	19	14	11	8	1	1	1

Proj. = Projection; Sel.= Selection.
Note 1: number of fully correct answers.
Note 2: number of correct answers from those correct at stage 1.

At the query translation stage, only 1 operation (not exist subquery) shows a significant drop in performance, compared to the easier operation. For the other operations, the p-values are not significant. This is a very surprising result. In most studies, projection and selection are treated as simple queries while operations such as join and repeated relations are treated as complex queries (Borthick et al., 2001; Bowen and Rohde, 2002; Chan et al., 1993; Smelcer, 1995; Jih et al., 1989; Welty, 1985). However, the results show that join, repeated relations and group+count operations have no significant effect at the query translation stage, which indicates that these operations are not difficult and subjects know what they want (data structure and operations) at the query translation stage.

At the query writing stage, the projection and selection operations are found to be simple operations. A query with just a projection operation received full grade, and a test for the selection operation (table 2, row 2) does not show any significant difference. The other operations, join, repeated relation, multiple joins, group+count and not_exist_subquery, are found to be difficult operations. Each of them significantly lowers performance at stage 2. The result is consistent with the literature reports on the difficulties of SQL. For example, Bowen and Rohde (2002) argued that end-users have difficulty with the join operation because they must reference additional tables in the FROM section. Referencing additional tables in the FROM section may also affect the names of the attributes listed in the SELECT section of the query, i.e., additional tables increase the likelihood that the SELECT attributes must be preceded by table qualifiers. Users must include additional WHERE JOIN statement to specify how to link the tables. Smelcer (1995) also found the SQL join statement to be particularly difficult for users.

Table 2. Non-parametric Wilcoxon Test Comparing Different Queries

Tested Operation	Data Comparison	Query Translation	Query Writing
Join	Q1 vs. Q2	z=-1.000(a) p=0.317	z=-2.236(b) p=0.025*

Selection	Q2 vs. Q3	z=-1.414(b) p=0.157	z=-1.000(b) p=0.317
Selection+ Repeated Relation	Q1 vs. Q4	z=-1.342(b) p=0.180	z=-3.317(b) p=0.001**
Multiple Joins	Q4 vs. Q5	z=-1.667(a) p=0.096	z=-2.121(a) p=0.034*
Group+Count	Q2 vs. Q6	z=0.000(c) p=1.000	z=-3.606(b) p=0.000**
Not_exist_subquery	Q2 vs. Q7	z=-2.000(b) p=0.046*	z=-3.742(b) p=0.000**
a: Based on negative ranks; b: Based on positive ranks; c: The sum of negative ranks equals the sum of positive ranks; * Significant at p<0.05; ** Significant at p<0.01.			

The exploratory findings are summarized as following:
1. The "projection" and "selection" operations have no significant difficulty impact on query performance during both stages. Subjects have little difficulty in understanding "projection" and "selection" operations and expressing them in SQL.
2. The "join", "repeated relations", "multiple join" and "group count" operations do not add any significant negative impact on performance at the query translation stage. But they do so at the query writing stage. The query writing stage results are consistent with literature findings, which also regard these operations as difficult. The query translation stage results are surprising, and contribute new knowledge that has not been studied before. The SQL syntax, and not the relational model itself, is the cause of the commonly found difficulties for these operations.
3. The "not-exist-subquery" operation has a significant negative effect on query performance at both stages. Subjects have difficulty both in understanding the concept of a not-exist-subquery and in expressing it in SQL.

Conclusion

This experiment adds a significant new perspective on database query research. It employs a cognitive theory of user query processing to empirically assess the effects of the data model (without a query language) and the additional impact of having to specify the query in a formal query language syntax. Prior studies on user query errors have concentrated on query accuracy at only the query writing stage. By conducting an experiment that measures query performance at both the query translation stage and the query writing stage, we can estimate the data model effect on user performance at stage 1, and the additional impact of a query language at stage 2. By comparing queries with different operations, and further comparing the same queries across stages, we are able to tell which operations of the data model are "inherently" difficult (caused by the model itself) and which operations are "additionally" difficult (caused by the particular query language syntax).

Findings at the query writing stage confirm findings in the literature about operation difficulties, thus providing strong support for the validity of this experiment. Findings at the query translation stage show surprisingly that some of the commonly perceived difficult operations are really not difficult at that stage. The results provide strong empirical support

that users can understand the relational model for many operations, but have difficulty expressing these operations in SQL.

This experiment contributes a new evaluation methodology that solves a longstanding research problem of how to apportion "blame" for query difficulties to the data model and the query language (Jih et al., 1989; Chan et al., 1993; Wu et al., 1994). This approach could be used in future comparison of user query performance across data models. Instead of comparing integrated packages (e.g., (data model X + query language Y) versus (data model A + query language B)), we could in future compare data models without any query languages (e.g. data model X versus data model Y), and further compare additional differences caused by query languages.

The practical implication of the research results is that before using relational database systems, users need more training on the particular difficulties of the query language, and also the operations that are even difficult at the model level (e.g. subquery with not exist). Knowing more about the difficulties that users have in expressing operations in SQL allows us to know which aspects of SQL cause problem for users and thus allow a more focused training for SQL users.

References

Androutsopoulos, I., Ritchie, G. and Tranisch, P. "Natural Language Interfaces to Databases－An Introduction." *Natural Language Engineering*, (1:1), 1995, pp.29-81.

Aversano, L., G. Canfora, A. De Lucia, S. Stefanucci. "Understanding SQL through Iconic Interfaces," *Proceedings 26th Annual International Computer Software and Applications Conference*, 2002, pp. 703-708.

Batini, C., Ceri, S. and S.B. Navathe, *Conceptual Database Design, An Entity-Relationship Approach*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1992.

Borthick, A.F., Bowen, P.L., D.R. Jones and M.H.K. Tse. "The Effects of Information Request Ambiguity and Construct Incongruence on Query Development," *Decision Support Systems*, (32:1), 2001, pp. 3-25.

Borthick, A.F., Bowen, P.L., S.T. Liew and F.H. Rohde. "The Effects of Normalization on End-user Query Errors: An Experimental Evaluation," *International Journal of Accounting Information Systems*, (2:4), 2001, pp. 195-221

Bowen, P.L., F.H. Rohde. "Further Evidence of the Effects of Normalization on End-user Query Errors: An Experimental Evaluation," *International Journal of Accounting Information Systems*, (3:4), 2002, pp. 255-290.

Bowen, P.L., O'Farrell, R.A., F.H. Rohde. "Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End-User Query Performance?" *25th International Conference on Information Systems*, 2004, pp. 141-156.

Chan, H.C., Wei, K.K. and Siau, K.L. "User-Database Interface: The Effect of Abstraction Levels on Query Performance," *MIS Quarterly*, (17:4), 1993, pp. 441-464.

Chan, H.C. and Wei, K.K. "Effect of Grading Schemes on Outcomes in Query Writing Experiments," *Interacting with Computer*, (8:1), 1996, pp. 7-12.

Chan, H.C., Tan, B.C.Y. and K.K. Wei. "Three Important Determinants of User Performance for Database Retrieval," *International Journal of Human-Computer Studies*, (51:5), 1999, pp. 895-918.

Chan, H.C., Teo, H.H., and Zeng, X.H. "An Evaluation of Novice End-User Computing Performance: Data Modeling, Query Writing and Comprehension." *Journal of the American Society for Information Science and Technology*, (56:8), 2005, pp. 843-853.

Connolly, T.M. and Begg, C.E. *Database Systems: A Practical Approach to Design, Implementation and Management*, 3rd ed. Addison-Wesley, 2002, pp. 67-194.

Codd, E.F. "Data Models in Database Management", *Proc. Workshop on Data Abstraction, Databases, and Conceptual Modeling*, 1980, pp. 112-114.

Date, C.J. *A Guide to the SQL Standard*, USA: Addison-Wesley, 1989.

Date, C.J. *The Database Relational Model: A Retrospective Review and Analysis*, USA: Addison-Wesley, 2001.

Galatescu, A. "A Unifying Translation of Natural Language Patterns to Object and Process Modeling," *Information Modeling in the New Millennium*, Idea Group Publishing, Rossi, M. and Siau, K. (Eds.), 2001, pp. 231-255.

George, D. and P. Mallery. *SPSS for Windows, Step by Step, A Simple Guide and Reference*, Allyn & Bacon, Needham Heights, MA., 2001.

Hutchins, E.L., Hollan, J.D. and D.A. Norman, "Direct Manipulation Interfaces," *Human-Computer Interaction*, 1, 1985, pp. 311-338.

Jih W.J. Kenny, D.A. Bradbard, C.A. Snyder, N.G.A. Thompson. "The Effects of Relational and Entity-Relationship Data Models on Query Performance of End-Users," *International Journal on Man-Machine Studies*, (31), 1989, pp. 257-267.

Kang, I.S., Bae, J.H.J. and Lee, J.H. "Database Semantics Representation for Natural Language Access," *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, IEEE, 2002, pp. 127-133.

Liao, C. and P.C. Palvia. "The Impact of Data Models and Task Complexity on End-User Performance: an Experimental Investigation", *International Journal of Human-Computer Studies*, (52:5), 2000, pp. 831-845.

Mannino, M.V. *Database Application Development and Design*, McGraw-Hill Company, Inc., 2001.

Mitrovic, A. "Learning SQL with a computerized tutor", *ACM SIGCSE Bulletin, Proceedings of the twenty-ninth SIGCSE technical symposium on Computer Science Education*, (30:1), 1998, pp. 307-311.

Ogden, W.C. "Implications of a Cognitive Model of Database Query: Comparison of a Natural Language, a Formal Language, and Direct Manipulation Interface", *ACM SIGCHI Bulletin*, (18:2), 1985, pp. 51-54.

Owei, V. "Natural Language Querying of Databases: An Information Extraction Approach in the Conceptual Query Language," *International Journal of Human-Computer Studies*, (53), 2000, pp. 439-492.

Owei, V., S. B. Navathe. "Enriching the Conceptual Basis for Query Formulation through Relationship Semantics in Databases," *Information Systems*, (26:6), 2001, pp. 445-475.

Qian, X. and Raschid, L. "Query Interoperation among Object-Oriented and Relational Dataases," *Proceeding of IEEE Data Engineering Conference*, 1995.

Ramakrishnan, R. and Gehrke, J. *Database Management Systems*, McGraw-Hill Companies, 2000.

Reisner, P. "Use of Psychological Experimentation as an Aid to Development of a Query Language," *IEEE Transactions on Software Engineering*, SE-3(3), 1977, pp. 218-229.

Reisner, P. "Human Factors Studies of Database Query Languages: A Survey and Assessment." *ACM Computing Surveys*. (13:1), 1981, pp. 13-31.

Rob, P. and Coronel, C. *Database Systems: Design, Implementation, and Management*, Fifth Edition, Course Technology, 2002.

Rosenthal, R. and R.L. Rosnow, *Essentials of Behavioral Research, Methods and Data Analysis*, 2nd edition, McGraw-Hill, Inc., 1992.

Shoval, P., and Shiran, S. "Entity-relationship and Object-oriented Data Modeling - An Experimental Comparison of Design Quality," *Data & Knowledge Engineering*, (21:3), 1997, 297-315

Siau, K. and X. Tan, "Cognitive Mapping Techniques for User-Database Interaction", *IEEE Transactions on Professional Communication* (49:2), 2006, pp. 96-108.

Siau, K., Y. Wand, I. Benbasat, "The Relative Importance of Structural Constraints and Surface Semantics in Information Modeling," *Information Systems*, (22:2/3), 1997, pp. 155-170.

Sinha, A. P., Vessey, I. "An Empirical Investigation of Entity-Based and Object-Oriented Data Modeling: A Development Life Cycle Approach." *Proceedings of the 20th International Conference on Information System*s, Atlanta, 1999, pp. 229-244.

Smelcer, J.B. "User Errors in Database Query Composition," *International Journal of Human-Computer Studies*, (42), 1995, pp. 353-381.

Srinivasan, A. and Irwin, G. "Data Abstractions and Their Use: An Experimental Study of User Productivity," *Human-Computer Interaction Interact '99*, IOS Press, Sasse, M.A. and Johnson, C. (Eds.), 1999, pp. 86-94.

Suh, K.S. and A.M. Jenkins. "A Comparison of Linear Keyword and Restricted Natural Language Database Interfaces for Novice Users", *Information Systems Research*, (3), 1992, pp. 252-272. 4.

Teo, H.H., Chan, H.C. and K.K. Wei, "Performance Effects of Formal Modeling Language Differences: A Combined Abstraction Level and Construct Complexity Analysis". *IEEE Transactions on Professional Communication*, (49:2), 2006, pp. 160-175.

Weber, R. "Are Attributes Entities? A Study of Database Designers' Memory Structures," *Information Systems Research*, (7:2), 1996, 137-162.

Welty, C. "Correcting User Errors in SQL," International Journal of Man-Machine Studies, (22), 1985, pp. 463-477.

Welty, C. and Stemple, D.W. "Human Factors Comparison of a Procedural and a Nonprocedural Query Language", *ACM Transactions on Database Systems*, (6:4), 1981.

Wong, C. and Luk, W.S. "Query Translation: The Query Interoperability Approach", *Multimedia, Knowledge-Based and Object-Oriented Databases*, Fong, J. and Siu, B. (Eds.), 1996, pp. 282-306.

Wu, C.Z., Chan, H.C., Teo, H.H. and Wei, K.K. "An Experimental Study of Object-Oriented Query Language and Relational Query Language for Novice Users," *Journal of Database Management*, (5:4), 1994, pp. 16-27.

Yen, M. and Scamell, R.W. (1993). "A Human Factors Experimental Comparison of SQL and QBE," *IEEE Transaction on Software Engineering*, (19:4), 1993, pp. 390-409.

Appendix: Database and Queries for the Experiment

This appendix contains the relational schema and the set of questions that were given to the subjects in the experiment.

1.1 Data Models

Employee (number, name, salary)
Engineer (number, profession)
Manager (number, rank)
Department (number, name, city)
Project (number, name)
Work (employee_number, department_number, date)
Management (manager_number, department_number, date)
Head (engineer_number, project_number, date)

Figure A1. The Relational Schema

1.2 Query Questions:

Q1: Show the department name and city.
Q2: Show the names of employees who head any project.
Q3: Show the names of employees who work in the sales department.
Q4: Show the names of employees with higher salaries than Jack's.
Q5: Show the names of employees who work in the same department as Jack.
Q6: List the names of managers who manage more than one department.
Q7: List the names of engineers who do not head any project.

1.3 Sample SQL Answers:

```
1. SELECT DEPARTMENT.NAME, DEPARTMENT.CITY
   FROM DEPARTMENT
2. SELECT EMPLOYEE.NAME
   FROM EMPLOYEE, HEAD
   WHERE EMPLOYEE.NUMBER=HEAD.ENGINEER_NUMBER
3. SELECT EMPLOYEE.NAME
   FROM EMPLOYEE, WORK, DEPARTMENT
   WHERE DEPARTMENT.NAME='SALES'
   AND WORK.EMPLOYEE_NUMBER=EMPLOYEE.NUMBER
   AND WORK.DEPARTMENT_NUMBER=DEPARTMENT.NUMBER
4. SELECT E1.NAME
   FROM EMPLOYEE E1, EMPLOYEE E2
   WHERE E1.SALARY>E2.SALARY
   AND E2.NAME='JACK'
5. SELECT E1.NAME
   FROM EMPLOYEE E1, EMPLOYEE E2, WORK W1, WORK W2
   WHERE E1.NUMBER=W1.EMPLOYEE_NUMBER
   AND E2.NUMBER=W2.EMPLOYEE_NUMBER
   AND E2.NAME='JACK'
   AND W2.DEPARTMENT_NUMBER=W1.DEPARTMENT_NUMBER
6. SELECT EMPLOYEE.NAME
   FROM EMPLOYEE
   WHERE EMPLOYEE.NUMBER=
```

```
   (SELECT MANAGEMENT.MANAGER_NUMBER
    FROM MANAGEMENT
    GROUP BY MANAGEMENT.MANAGER_NUMBER
    HAVING COUNT (*)>1)
7. SELECT EMPLOYEE.NAME
 FROM EMPLOYEE
 WHERE EMPLOYEE.NUMBER=
   (SELECT ENGINEER.NUMBER
    FROM ENGINEER
    WHERE NOT EXIST
      (SELECT *
       FROM HEAD
       WHERE HEAD.ENGINEER_NUMBER= ENGINEER.NUMBER) )
```